

[USS][Desconflito] Código Exemplo

Exemplo de código Provedor de Desconflito (Go e Python): https://github.com/dp-icea/scd_provider

Para realizar autenticação:

1. Verificar se está gerando o token

Para criar OIR:

1. Enviar GeoJSON da área desejada

Abaixo segue exemplo de código em Python:

app.py

```
from flask import Flask, request

from dss import Dss

import json

app = Flask(__name__)

database = {}

dss = Dss()

@app.route('/uss/v1/operational_intents/<operational_intent_id>', methods=['GET'])
def get_oir(operational_intent_id):
    print(database)
```

```
if operational_intent_id not in database:
```

```
    return {"msg": 'No such OIR'}, 404
```

```
return json.dumps(database[operational_intent_id], default=lambda o: o.__dict__)
```

```
@app.route('/injection/oir', methods=['PUT'])
```

```
def inject_oir():
```

```
    volume = request.get_json()
```

```
    try:
```

```
        dss.conflict_manager.check_restrictions(volume)
```

```
        dss.scd.check_strategic_conflicts(volume)
```

```
        oir = {}
```

```
        oir["operational_intent"] = {}
```

```
        oir["operational_intent"]["reference"] = dss.scd.put_operational_intent(volume)
```

```
        oir["operational_intent"]["details"] = {
```

```
            "volumes": [],
```

```
            "off_nominal_volumes": [],
```

```
            "priority": 0
```

```
        }
```

```
        oir["operational_intent"]["details"]["volumes"].append(volume)
```

```
        database[oir["operational_intent"]["reference"]['id']] = oir
```

```
except Exception as ex:
```

```
    print(f"Erro na criação: {ex}")
```

```
    return {"msg": "Erro"}, 400
```

```
return {"Success": True}, 201
```

```
if __name__ == '__main__':
```

```
    app.run(port=5050)
```

dss.py

```
from conflict_manager import ConflictManager
from scd import Scd
```

```
class Dss:
```

```
    def __init__(self) -> None:
        self.conflict_manager = ConflictManager()
        self.scd = Scd()
```

scd.py

```
import requests
```

```
import uuid
```

```
from env import USS_BASE_URL, DSS_HOST
```

```
class Scd:
```

```
    def __init__(self) -> None:
        self.auth()
```

```
    def auth(self):
        url =
        "http://kong.icea.decea.mil.br:64235/token?grant_type=client_credentials&intended_audience=localhost&issuer=localhost&scope={0}"
        self.strategic_coordination =
        requests.get(url.format("utm.strategic_coordination")).json()["access_token"]
```

```
    def check_strategic_conflicts(self, volume):
```

```
        url = DSS_HOST + "/dss/v1/operational_intent_references/query"
        body = {
```

```

        "area_of_interest": volume
    }

    header = {"authorization": f"Bearer {self.strategic_coordination}"}
    response = requests.post(url, headers=header, json=body).json()

    if (len(response['operational_intent_references']) > 0):
        raise Exception(f"Interseção com outra Intenção")
    {response['operational_intent_references'][0]['id']}")
    else:
        print("Sem intenções para o volume")

def put_operational_intent(self, volume):
    id = str(uuid.uuid4())
    url = DSS_HOST + f"/dss/v1/operational_intent_references/{id}"

    body = {
        "flight_type": "VLOS",
        "extents": [volume],
        "key": [],
        "state": "Accepted",
        "uss_base_url": USS_BASE_URL,
        "new_subscription": {
            "uss_base_url": USS_BASE_URL,
            "notify_for_constraint": False
        }
    }

    print(body)

    header = {"authorization": f"Bearer {self.strategic_coordination}"}

    response = requests.put(url, headers=header, json=body).json()

    print(f"OIR criada com id: {id}")
    print(response)

    return response['operational_intent_reference']

```

conflict_manager.py

```
import requests
import uuid

from env import USS_BASE_URL, DSS_HOST

class Scd:

    def __init__(self) -> None:
        self.auth()

    def auth(self):
        url =
"http://kong.icea.decea.mil.br:64235/token?grant_type=client_credentials&intended_audience=localhost&is
suer=localhost&scope={0}"
        self.strategic_coordination =
requests.get(url.format("utm.strategic_coordination")).json()["access_token"]

    def check_strategic_conflicts(self, volume):
        url = DSS_HOST + "/dss/v1/operational_intent_references/query"
        body = {
            "area_of_interest": volume
        }
        header = {"authorization": f"Bearer {self.strategic_coordination}"}
        response = requests.post(url, headers=header, json=body).json()

        if (len(response['operational_intent_references']) > 0):
            raise Exception(f"Interseção com outra Intenção
{response['operational_intent_references'][0]['id']}")
        else:
            print("Sem intenções para o volume")

    def put_operational_intent(self, volume):
        id = str(uuid.uuid4())
```

```
url = DSS_HOST + f"/dss/v1/operational_intent_references/{id}"

body = {
    "flight_type": "VLOS",
    "extents": [volume],
    "key": [],
    "state": "Accepted",
    "uss_base_url": USS_BASE_URL,
    "new_subscription": {
        "uss_base_url": USS_BASE_URL,
        "notify_for_constraint": False
    }
}

print(body)

header = {"authorization": f"Bearer {self.strategic_coordination}"}

response = requests.put(url, headers=header, json=body).json()

print(f"OIR criada com id: {id}")
print(response)

return response['operational_intent_reference']
```

Revision #5

Created 2 July 2024 11:40:55 by Cenato

Updated 2 July 2024 12:04:07 by Cenato